Logo in Education: What, How, Where, Why and Consequences

Planning Services

Alberta
EDUCATION

Digitized by the Internet Archive
in 2015

https://archive.org/details/logoineducationw00kier

APR 1

Alberta
EDUCATION

Devonian Building, West Tower, 11160 Jasper Avenue, Edmonton, Alberta, Canada T5K 0L2

February, 1985

To The Reader:

This report on the computer language Logo is a study designed to alert the reader to the wide application of Logo to all grade levels, in all subjects, and to all types of students.

As the author notes, regular use of Logo may enhance general thinking skills, a variety of mathematical thinking skills and some problem solving skills. Hence this study of Logo certainly warrants our interest.

Alberta Education is pleased to provide this copy of Logo in Education: What, How, Where, Why and Consequences.

Sincerely,

H. I. Hastings

H.I. Hastings
Director
Planning Services

LOGO in Education: What, How, Where, Why and Consequences

Thomas E. Kieren
University of Alberta

December 1984

Funding Provided by Planning Services Branch,
Alberta Education

TABLE OF CONTENTS

1. Introduction....................1

2. The Instructional Structure of Logo
(Aspects of Logo, Strucharts, Levels, Debugging,
Logo and Language Use, Ego Syntonic Mathematics)....................2

3. Logo in Use
(Visions of Logo Use, Structures of Logo Use,
Computer Literacy/Computer Science, Mathematics, Art,
Language, Data Management, Social Studies, Music,
Problem Solving, Logo and Less Able Students)....................18

4. Why Logo: The Word From Research
(Effects, Language Use, Teacher Education, Cast)....................29

5. Recommendations....................34

1. INTRODUCTION

This report is the product of a number of different considerations. An extensive body of literature was perused. While some documents date back 15 years or so, most literature pertaining to Logo postdates Papert's Mindstorms (1980). A product of this part of the study is found in the first two appendices.[1]

In addition, I met with the following persons: Dr. Marvin Westrom; Mr. William Egnatoff (UBC), Ms. Briget Knoepfell, Mr. Steve Williams (Richmond, B.C.), Dr. George Cathcart, Mrs. Gloria Cathcart, Dr. Al Olson, Mrs. Pat Richardson, Ms. Judy Dobson, Dr. Daiyo Sawada, Dr. D. B. Harrison, Dr. Tom Schroder, Dr. Margret Moore, Dr. Bill Burger (Oregon State University), Dr. Gary Bitter (Arizona State University), Dr. Dale Burnett, Dr. Wm. Higginson (Queens), Dr. Carolyn Kieran (University of Quebec at Montreal), Dr. Alan Osborne and Dr. Pat Wilson (Ohio State University), Dr. L. L. Hatfield (University of Georgia), Dr. Catherine Berdonneau (Paris, France), Dr. W. Lordon (Witwatersrand University, South Africa), and B. Simon (Sydney, Australia). Where information in this report arises from these conversations, this will be indicated by (1984*). All of these persons are directly involved with Logo projects in schools and were familiar with projects in their regions and countries as well as literature of interest.

The literature review and consultations form the basis for this report, which attempts to give answers to the following questions:

1. What is Logo?

---

[1] Mr. Peter Wood of the Department of Secondary Education, U of A, assisted with literature/material review.

2. How is Logo used?
3. In which subject areas or aspects of curriculum/instruction is Logo used?
4. From the limited research literature, why might Logo be used/not used? What are the effects of different kinds of Logo use?
5. What recommendations for Logo use in Alberta arise as consequences from the answers to the previous questions?

Of course, it must be noted that Logo is currently being used in a variety of classrooms in a variety of ways in elementary, secondary and faculty of education classrooms in Alberta. Thus, this report provides perspectives for viewing, evaluating and perhaps formalizing and improving such use.

2. The Instructional Structures of Logo:

The structures behind the "song of the turtle."

Logo, a primarily symbolic rather than quantitative language, was developed through work at the firm of Bolt, Baranak and Newman and at Massachusetts Institute of Technology (MIT) in the late 1960s in Boston. Its initial school applications were linguistic in nature, relating to Logo's list processing LISP-like character. The Artificial Intelligence Laboratory at MIT established a Logo group in the late 1960s and work continues there today. Although many names are associated with the development of Logo (Feurzig, Abelson, deSessa, Watt, Lawler, Thornburg, Ross, Moore, Howe, Higginson and Burnett to name a few), over the last 10 years, it has been Seymour Papert who has been its key developer, philosopher and spokesperson. One might speculate, aside from Papert's personal drive, as to why MIT would focus on Logo. Perhaps in studying artificial intelligence, it is useful to devise ways in which naive human intellect can interact with an artificially intelligent device/ environment. Logo is such a way.

What is the nature of Logo? There are sources which discuss Logo as a virtual language and the linguistics of Logo as a computer language. Here we look at the nature/structure of Logo as it appears in an instructional environment. It is critical to this discussion to think of Logo as a means of controlling and inventing robotic phenomena. In addition to driving the well-known "Turtle," both the floor robot and the turtle of the two dimensional screen world, Logo can be used to generate letter, story and poetry makers, to drive various musical devices as well as art making devices. For the sake of brevity, the discussion of the instructional structure of Logo will centre around its "turtle" mode.

There are two other related aspects of Logo which also bear on the following discussions. First, Logo is a list processing language and handles all communications as lists. As Allan (1984) notes, "Logo without lists is not Logo, but merely another graphics system." So although our discussion centres around the "turtle" mode, the instructional structure described below and its orientations and processes are best understood in the light of Logo's list processing nature.

A second aspect of Logo, which bears particularly on the microcomputer versions of Logo, is that Logo is fully recursive. This feature will be discussed below, but it is important to note that Logo procedures are executed at several logical levels which means that certain Logo procedures, even simple ones involving the REPEAT command, are very memory intensive. Much microcomputer Logo has been done on 64K machines, which is a minimal requirement. It is likely that Logo power will be enhanced through the use of machines with larger and better structured memories.

In order to understand the implications of Logo for computer literacy, problem solving and subject matter implementations, it is important to analyze the structure of Logo. Figure 1 is a representation of the instructional structure of Logo. It is the basis for the following discussion and is useful in interpreting the research and experiential findings discussed later.

2.1 Word

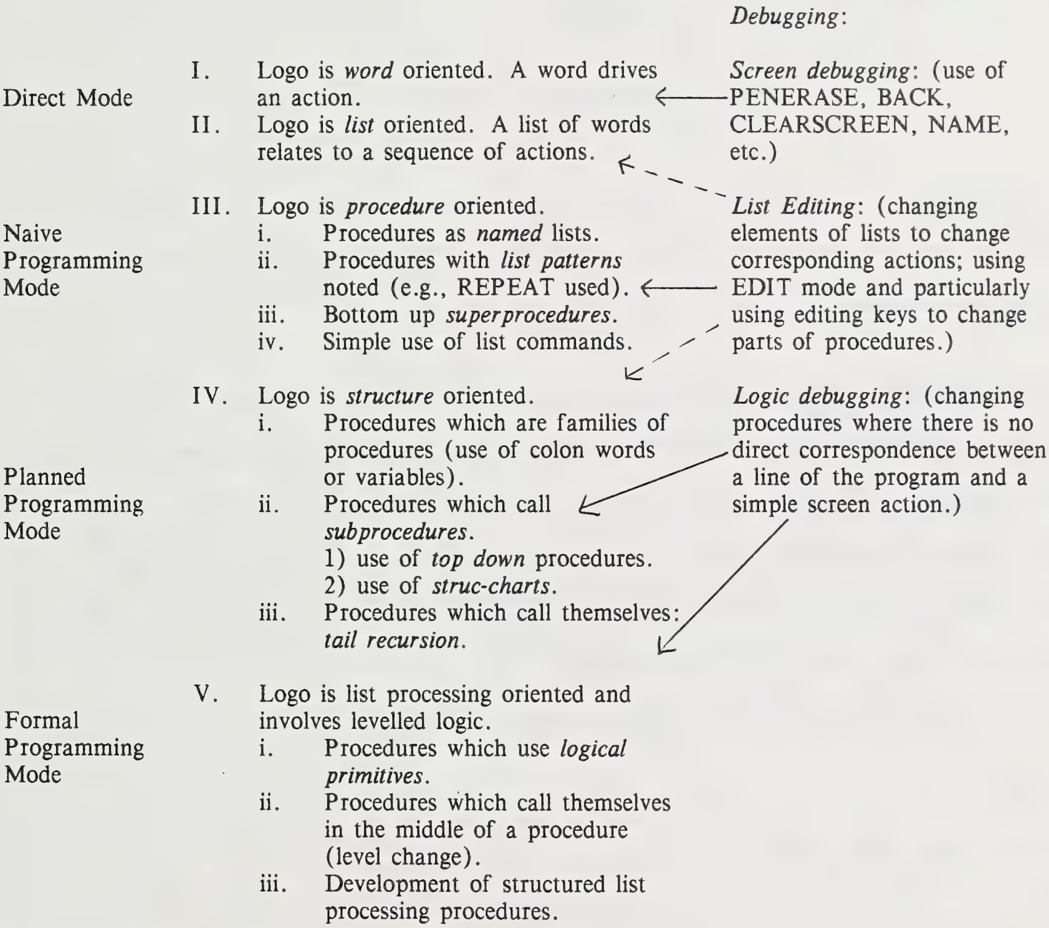
Because Logo is an interpreted language, giving a command word which is "known" by the turtle results in a related screen action. These words can be primitives of Logo (e.g., FORWARD 100, RIGHT 270, PC 4) or command words which have been "taught" to the turtle--procedures named by a single command word, written by the users, and loaded into the working memory of the computer. Giving a command word (or a list of such words) and pressing the "return" provides a screen (floor turtle or other robotic device) response. This is called functioning in the direct mode.

2.2 List

A complex pattern in direct mode is accomplished by giving a list of words, each followed by a return or a list followed by a return. The key actions for students here are:

- given a pattern, recognize or generate a list of words to get it.
- given a list, execute it and/or predict its outcome beforehand.

This kind of Logo use allows one to generate (and save, using SAVEPICT in Terrapin Logo, for example) complex products. (Such products can also be printed out in hard copy or used in any procedure in other languages, e.g., BASIC or PASCAL as binary files).



Debugging:

Direct Mode

I. Logo is *word* oriented. A word drives an action.

II. Logo is *list* oriented. A list of words relates to a sequence of actions.

Screen debugging: (use of PENERASE, BACK, CLEARSCREEN, NAME, etc.)

Naive Programming Mode

III. Logo is *procedure* oriented.
  - i. Procedures as *named* lists.
  - ii. Procedures with *list patterns* noted (e.g., REPEAT used).
  - iii. Bottom up *superprocedures*.
  - iv. Simple use of list commands.

List Editing: (changing elements of lists to change corresponding actions; using EDIT mode and particularly using editing keys to change parts of procedures.)

Planned Programming Mode

IV. Logo is *structure* oriented.
  - i. Procedures which are families of procedures (use of colon words or variables).
  - ii. Procedures which call *subprocedures*.
    1) use of *top down* procedures.
    2) use of *struc-charts*.
  - iii. Procedures which call themselves: *tail recursion*.

Logic debugging: (changing procedures where there is no direct correspondence between a line of the program and a simple screen action.)

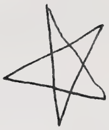
Formal Programming Mode

V. Logo is list processing oriented and involves levelled logic.
  - i. Procedures which use *logical primitives*.
  - ii. Procedures which call themselves in the middle of a procedure (level change).
  - iii. Development of structured list processing procedures.

Figure 1
An Instructional Structure of Logo

List pattern

```
FD  20 [RETURN]
RT 144 [RETURN]

FD  20 [RETURN]
RT 144 [RETURN]

FD  20 [RETURN]
RT 144 [RETURN]

FD  20 [RETURN]
RT 144 [RETURN]

FD  20 [RETURN]
RT 144 [RETURN]
```



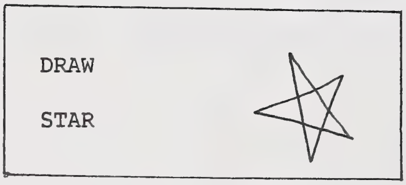
The above list generates a pentagram or familiar star. Other good list exercises are generating or guiding the turtle (floor or screen) through mazes or obstacle courses.

2.3 Procedures

Logo has DEFINE and EDIT modes which allow one to generate new words whose definition a robotic device can follow or act out. For example, a TO STAR command followed by the list above, would generate a word STAR which could then be executed by the turtle (and saved to disk). (TO followed by a single word puts Logo into define mode).

One could edit such a procedure, noting the <u>list pattern</u>.

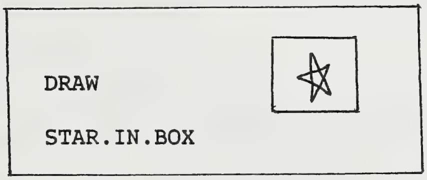
```
 TO STAR
>REPEAT 5 [FD 20 RT 144]
>END
```



```
DRAW
STAR
```

If one had a procedure BOX, one could write a superprocedure, TO STAR.IN.BOX

```
>STAR

>PU LT 90 FD 40

>LT 90 FD 20 RT 180

>PD

>BOX

>END
```

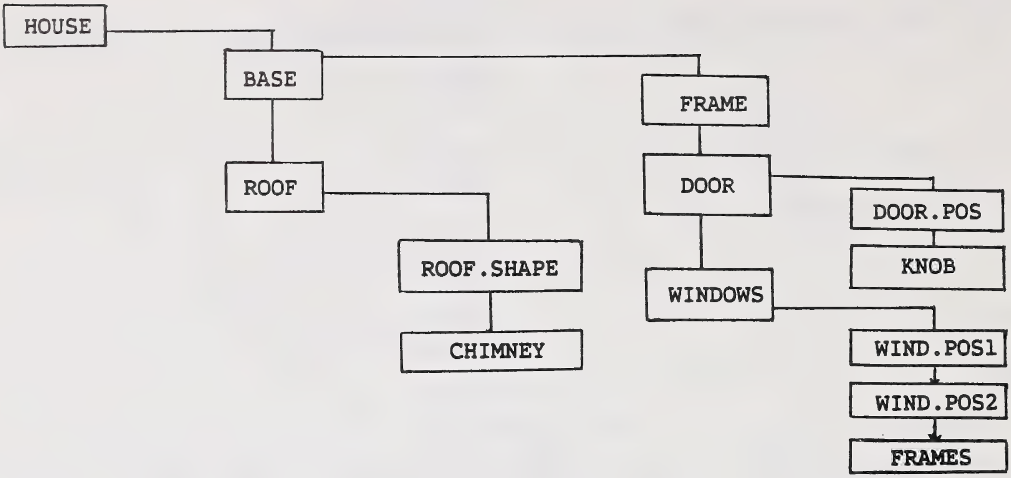


Such procedure patterns and superprocedures represent a naive, exploratory and rudimentary programming mode in Logo.

2.4 Structure

Another level of Logo use is represented by the development and use of structured procedures. These procedures are pre-planned.

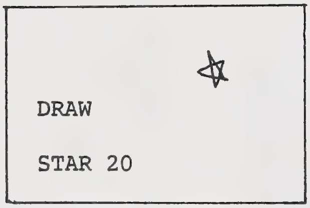
Such plans can be represented in struc-charts.



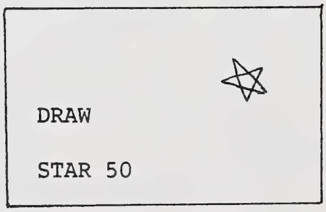
Such a chart does not necessarily trace the flow of action directly, but it does show the relationships or structure among the words which will make up the procedure. Further, it shows words or lists which must be developed to fully define the procedure HOUSE.

A simpler kind of structured program is one which recognizes that a shape comes in numerous sizes. Thus, for example, by changing all of the sizes appropriately without changing the angles in a procedure, one can generate a family of shapes. One uses "colon words" to denote variables.

```
TO STAR :SIZE
>REPEAT 5 [FD :SIZE RT 144]
>END
```



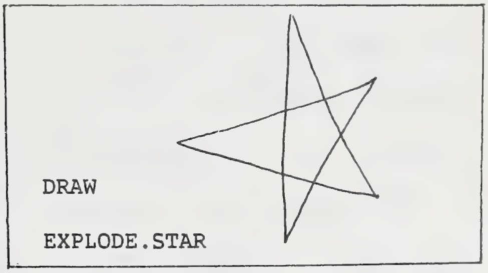
```
DRAW
STAR 20
```



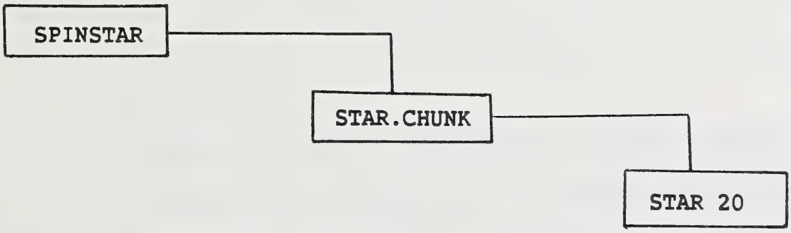
```
DRAW
STAR 50
```

One can use such a procedure to build others.

```
TO EXPLODE.STAR
>STAR 5
>STAR 20
>STAR 80
>END
```

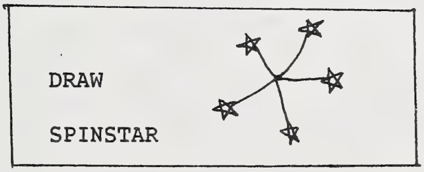


```
DRAW
EXPLODE.STAR
```



```
TO SPINSTAR
REPEAT 5 [STARCHUNK RT 72]
END
```

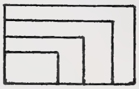
```
TO STARCHUNK
>FD 20
>STAR 20
>BK 20
```



```
DRAW
SPINSTAR
```

Finally, some simply structured procedures call themselves at the end (and usually involve logical primitives to stop). For example:

```
TO GROWBOX :SIZE

>IF :SIZE > 50 THEN STOP

>BOX :SIZE

>GROWBOX :SIZE + 10

>END
```



DRAW

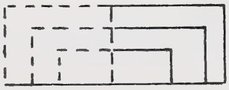
GROWBOX 20

While this use of recursion seems to provide Logo with a "looping" structure, the logic is considerably more subtle. Each call of GROWBOX involves another logical level of execution.

2.5 Logo Logic

There are numerous logical and list processing primitives available in various micro Logos. One example of a more sophisticated Logo logic involves embedded recursion.

```
TO REFLECT.BOX :SIZE

IF :SIZE < 1 THEN STOP

REPEAT 4[FD :SIZE RT 90]

REFLECT.BOX :SIZE-10

PC 4

REPEAT 4[FD :SIZE LT 90]

END
```



DRAW

REFLECT.BOX 30

In this procedure levels 30, then 20 then 10 all draw appropriate boxes (shown on right). When level 0 is reached, the program stops at that level and returns for unfinished business at level 10 (change color--indicated by dashes--and draw the left box of size 10). Level 10 stops and returns control to level 20 and so on. There is no direct way from reading the procedure that one could follow the action on the screen. This is purely a logical procedure with a logical, but visible outcome.

2.6 Levels

As suggested by the instructional diagram, there are definite levels of Logo use. There are two direct levels or immediate mode levels where there is direct word-action feedback. Depending on the words available or the designed tasks, one can do sophisticated, but not formal, activities at these levels.

The naive programming stage represents a rather simple first programming level. Again, through the use of REPEAT and superprocedures, rather spectacular projects are available at this level.

The latter two levels represent more insightful programming. These levels demand the ability to make at least simple concrete generalizations and to pre-think the structure of the task. Although these structural thinking experiments can be kept simple, the teacher's word (for example, "use subprocedures" as discussed in Mindstorms, Papert [1980]) is likely to be, for some children (and young and old adults), more a call to follow the "teacher's way" rather than a natural thinking act. Recursion, even tail recursion, but certainly embedded recursion, calls for a logical structural analysis as well as a task or figural structural analysis.

One consequence of the levelled structure of Logo is a call to proceed "slowly" when instructing students in Logo use. Better to do sophisticated, successful things at a lower level, than simple but rather rote things at a higher level of language use. This also suggests that learning LOGO at any level is not the producct of a short workshop experience, but is likely to be a longer time proposition to be effective (Krasner & Mitterer, 1984).

2.7 On Debugging

A central theme of Mindstorms (Papert, 1980) is that having a young person program a computer puts that person in active contact with powerful general thinking tools. One such tool is debugging. An algorithm or procedures which does not work, or works in ways not anticipated, is not considered a failure for the programmer, but as a challenge to alter the faulty procedure. Such alteration of thinking and procedures gives one a sense of successively approximating an ideal solution and hopefully has the mathemagenic effect of making one's future procedures more clear and structured.

While debugging occurs under any computer program-making regime, the Logo instructional structure seems to foster its appropriate use. As indicated in the diagram, debugging in Logo occurs at several levels--three of which have been identified. At the first level, the user executes a single command or list of commands and finds that the effect on the screen is wrong or not as hoped. With the cognitive ability to see an undoing action for a given action, one which generally is developed rather early in the school years, a young user can debug at a visual screen level. In its least developed form, this may mean clearing the screen and starting over again. However, one should quickly learn to "erase" just the wrong parts of the product and execute commands which will appropriately replace them. (For younger users the PENERASE command in Apple Logo is easier to grasp than that of setting the pen color to match the background as is done in some other Logo versions).

A second level of debugging involves list editing. Here its user changes elements in a list or more importantly, in Logo uses the EDIT mode to alter procedures. A key computer literacy outcome here is the learning of an editing system (a set of keys which allow one to move the cursor and change lists in various ways). Of course, just knowing such a system is not a sufficient condition for this kind of debugging. One needs to be able to come to relate a segment of the procedure with a part of the sequence of actions which is driven by the procedure. This requires one to think of a list of commands or words at a distance from the associated acts. Logo proponents, including Papert (1979, 1980), claim that such debugging fosters more structural procedure planning so as to more easily relate subprocedures to action parts, and isolate errors for easier identification. Such thinking is important in modern computer programming and analysis work at any level.

The most sophisticated kind of debugging, which is an extension of that above and different from it in that it is more formal, is logic debugging. A key feature here is that a person is able, at a purely language/logic level, to trace errors. As procedures become more complex and at a higher level, there is less direct correspondence between small procedure pieces and the resulting product. This kind of debugging can take place from an analysis of a struc-chart without ever considering the result of executing a procedure. Thus, the thinking behind much debugging is more formal in character and perhaps most likely to transfer or be related to thinking in non-computer domains. It is also the most analytic and likely to occur with young persons at a more advanced developmental stage (Krasner & Mitterer, 1984), with longer Logo experience, including experience with more sophisticated aspects of Logo.

The consequences of this analysis are rather obvious. The rudiments of the thinking tool "debugging" are available at the earliest levels of Logo use. Thus, anyone using Logo can be expected to debug his or her computer work.[2] However, because the first two levels of debugging are very centred on the machine environment, the likelihood of transfering the thinking patterns to other environments seems most likely to occur from logic debugging. Thus, more extended Logo use, both in time for an individual, and through a larger age range, at least to upper elementary school and probably junior high school, seems important if the general benefits of this thinking skill are to be realized.

2.8 On Logo as a Language

The eminent Canadian language scholar Northrup Frye (1982) has noted that language use has developed historically through several levels. The two earliest levels of use which he terms hieroglyphic and hieratic relate to an immanent connection between language and object or action. In the former level, a word is only used in association with a [visible] object. In hieratic use, a word is "put for" an object or calls forth an object. In relating this to Logo language use one can see that the direct mode represents a very immanent use of language. A command word is either seen as directly related to a robotic action or as immanently calling such action.[3]

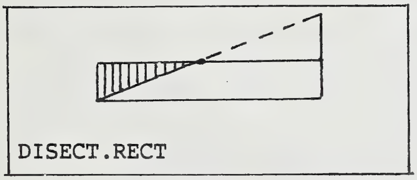
Another of Frye's historic language use levels is what he terms metonymic. Here language is used as an analogy for actions or objects. At this level, language can be manipulated independently of the world of action, but the user is cognizant of an action change analogous to language change. In Logo use, the writing of especially simple, but also structured, procedures represents such an analogous use of language. Because Logo is word and structure oriented, a junior high class could use a procedure as follows to justify the formula relating the area of a triangle to a rectangle.

---

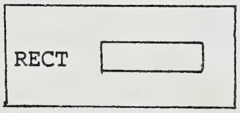
[2] It should be noted that some very young children, K and early primary, need considerable teacher support to do this kind of debugging (Burnett, Higginson [1984*], Egnatoff, Knoepfell [1984*]).

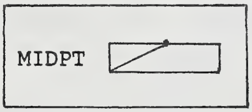
[3] This is vividly seen in the fact that while the "20" in FD 20 is easily understood by division one students, the "20" in RT 20 initially causes consternation. It seems that the word "right" is a metaphor for an orthogonal turn for most young children.

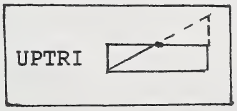
```
TO DISECT.RECT

>RECTANGLE

>MIDPT

>UP.TRIANGLE

>ER.TRIANGLE

>END
```

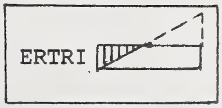


Because of the structured nature of Logo, one could execute it as a whole--the procedure "drives" the geometric result (a proof in a sense). One also could "look inside" the procedure with a class and execute each piece.









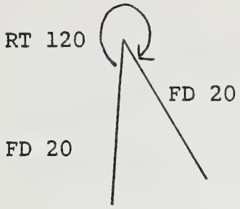
This "animated" sequential use of the commands of the procedure clearly show the Logo procedure in whole and part as analogous to action. It also suggests a value of structure in programs in general.

Finally, Frye suggests that language can be used <u>demotically</u> or <u>analytically</u>. Here, language is used without necessary reference to action or object. Although in Logo, executing procedures has a robotic referent or effect, procedures can be written and changed for their own sake. Further, in some procedures there is no obvious connection between a procedure piece and any associated action. One's analysis of the procedure would be at an analytic level, as would its development.

Because of its direct mode aspect, Logo can be used as a language in an immanent way by students to accomplish a complex task in a word by word fashion. Because of its procedural and structural nature, Logo is most often used analogically. This kind of language use is very effective in studying the logic of action. Of course, computer programs in any language can be thought of as language used in this fashion. However, in a language like BASIC the part by part analogy would be more difficult to observe because a person's BASIC procedure is not necessarily structured. On the other hand, while Pascal is structured, because of its technical nature it is not easy to use by naive users.

2.9 Turtle Geometry--Ego Syntonic?

In order to understand the brief discussion of the use of Logo in mathematics below, it is necessary to interpret one of Papert's (1980) more diffcult concepts--ego-syntonicity. Papert (1980) talks of understanding procedures in Turtle geometry by "playing turtle" or acting them out. Papert claims that the simpler aspects of Turtle geometry are body syntonic in this way--the "turtle's" action is in sync with a person's action. In claiming that turtle geometry is ego syntonic, he is claiming that its geometry is in sync with the way in which a person would build up geometric ideas. From a mathematical point of view, Turtle Geometry is a local geometry--seen and done from an internal, personal point of view. Thus, an angle is looked at from the amount of turn used in drawing it, rather than as a product, or a pair of segments separated.



Beyond being personally constructive, Kieren and Olson (1983) have argued that Turtle Geometry can match the sequential levels of geometric development as discussed by the van Hieles (1980). These levels successively treat objects as (0) wholes, (1) bearers of properties, (2) related logically, and finally as (3) deducible from an axiomatic system. One can study objects as wholes or property bearers in direct mode of Logo. One considers property sets and logical relationships among objects in the naive programming mode (simple procedures) and in the structured programming mode. Although it is not as clear, one could envision Logo tasks which emphasized axiomatic micro-worlds and hence would be level 3 geometry. From developmental pedagogical studies, the van Hieles have found that instruction which is above the level of the student leads at best to rote (non-ego syntonic) geometry.

Another aspect of ego syntonicity is manipulation of imagery in one's mind. Logo in language and robotic action allows one to test one's mental manipulation against a screen manipulation. Further, one can trace the consequences of such mental imagining by altering one's Logo procedures.

Two consequences of this analysis are as follows. First, Logo geometric tasks can be devised for students who are at any of the levels 0, 1 and 2 and perhaps 3. Thus, turtle geometry is effective for elementary school children. Yet asking these children to do structured geometric procedures may lead them to be programming only in an imitative way. Various experiences with children here in Canada (Olson [1984*], Kieran [1984*], Moore [1984*], Berdonneau [1984*], Lorden [1984*], Noss [1984]), this continent and elsewhere seems to indicate that younger (under 10 or 11) and naive users of Logo of any age, do not naturally do a structural analysis of a program. Similarly, although such users can imitate the use of recursion, they will usually choose to use other means in procedures they write.

2.10 Summary

The purpose of the above analysis has been to give several ideas on the instructional structure of Logo, its inspiration of the use of thinking tools such as debugging, the nature of language use under Logo, and the relationship of computer use under Logo to the study of geometry, one possible subject application.

3. Logo in Use

3.1 How and WHO

The literature, discussion with Logo experts and site visitations suggest that Logo is being used today mainly by two populations--elementary school children and adults (primarily teachers or teachers in training). However, the weight of this empirical evidence should not lead one to believe that this is a necessary condition. There are a growing number of places in Alberta and elsewhere where Logo is used with junior high school students (Olson [1984*], Harrison [1984*]). It has been used in senior high (Wood, 1984) and university mathematics courses (e.g., Oregon State; Burger, 1984*). It is part of a computer science sequence at Oregon State (Moore, 1984*) and forms the core of a sophisticated mathematics course--"turtle geometry" at MIT (Abelson & DiSessa, 1981) and elsewhere.

3.2. On Visions of Logo Use

There seem to be several camps of use in Logo. For example, Bitter (1982-83), in his program of computer literacy development, sees Logo being used with very young children as a basis for computer literacy. His suggested curriculum abandons Logo explicitly after Grade 3. This seems to fly in the face of experience with Logo in projects here in Edmonton and Calgary, as well as the generative work at MIT. Bitter (1984*) himself suggests the value of continuing Logo use but also notes that there are many other aspects of computer use which are better accomplished with other languages and software (e.g., numerical problems and business applications).

A second use style is Logo for literacy, but also as a basis for computing science. In this use, the notions and nature of robotic actions, and computer science notions such as procedure, word uses (command words, variable words, words per se), structure and recursion are illustrated in sets of tasks. These tasks might be worked on by whole classes of elementary school students (e.g., Campbell [1984*] in Richmond, B.C.), enrichment groups (Knoepfell [1984*] in B.C.), groups of secondary school students with learning difficulties (e.g., Wood [1984] Edmonton), or even by groups of teachers (Ferres [1983], Oregon), or a computing science class (Moore [1984*], Oregon State). Perhaps the most eloquent discussion of Logo and computer science is found in Allan (1984). He emphasizes the list processing nature of Logo, how words are used in such a system, the local variable use in Logo, and how Logo structures computer use.

Another kind of use emphasizes Logo as a generator of problem solving activity. Within this, there are those who emphasize child generated general problem solving through Logo (e.g., Dobson & Richardson [1984]), and those who use Logo approaches to have young persons solve pre-set interesting mathematical, computer or graphical problems (e.g., Gloria Cathcart [1984*]).

A fourth camp sees applying Logo in studying other subject areas. Some of these uses focus on Logo programs as utilities for studying mathematical (Therrien, 1984), language (Dewitt, 1984), musical (Bamberger, 1982) or geographical (Thornburg, 1983) structures. Others see the actual writing of procedures as central to developing ideas.

A fifth group take the Papert (1979, 1980) concept of exploratory micro-worlds as central to their Logo use. This is particularly seen in some of the original MIT applications (Lawler [1976], [1983]; Burnett [1981]), in some work at Lamplighter School and in various interesting pieces of personal work done by students or even adults (e.g., Burnett [1984*] on Islamic art or model railroading in Logo).

3.3 On Structure in Logo Use

There are many levels of structure or ways of organizing Logo use. At the extreme of structured use is the selection of a number of Logo tasks, each of which is in the form of a rigid set of steps to follow, all done by all students (perhaps as part of a computer literacy unit). This use seems to contradict Logo philosophy and seems to miss the point of the instructional structure and language use structure of Logo. Thus, though this kind of use has been tried, it is not the recommended structure.

Perhaps the most common way of organizing instruction in Logo is through the use of various task sheets. These tasks (and related Logo, mathematical, graphical, musical, etc., material) are frequently posted on large posters in the computer room or area of the classroom. For example, a project in South Africa (Lorden, 1984*) uses cards which give users (aged 10-14) play shapes (pre-written procedures which students can "play" with in direct mode, black box tasks where a student, through use, analyzes the nature of a Logo procedure, and work tasks which are structured, but open direct or program mode tasks matched to the level of students (age and/or experience). Recent cards from Access (Richards, 1984) fit this latter category or a category mentioned previously--open problems. Higginson and Burnett (1984*) see the most common use of Logo by intermediate and early junior high students to be a combination of card use, followed by student's own related extensions. Dobson and Richardson (1984*) and Cathcart and Cathcart (1984*) report the same phenomenon.

All of the above organizational patterns can be developed so that one can match tasks with particular objectives of computer literacy, science, a particular subject area, or of particular aspects of problem solving. Developing various Logo micro-worlds represents perhaps the most open Logo use structure. Here, the learner is given primitives or developed procedures relating to a particular artificial or real robotic environment. The user is then free to develop uses and further procedures in his/her own way.

In all but the first pattern suggested, students are likely to be working on a wide variety or level of problems in the same class. Because this places the teacher in more of an advisory role rather than a "telling" role, such variance is seen as having merit by many (e.g., Noss [1984], Berdonneau & Dumas [1981]).

Nearly all of the Logo teachers seem to have students working in groups of two or more, or at least in situations where they can and do readily communicate with one another. Kieran (1984*), Burnett (1984*), and Noss (1984) all stress the value of such groups but maintain that two is the optimal functional group size.

In its initial inception, Logo was designed as a powerful computer language environment accessible to young persons, particularly children. While it has certainly been used as a vehicle for computer literacy for children, it is and has been used for many other purposes as well. What follows is an abbreviated catalogue/description of such uses.

3.4 Computer Literacy/Computer Science

Logo has been viewed by some as a child's first step toward computer literacy. Bitter (1982-83), in his scope and sequence, sees Logo as the language of choice in grades 1-3. While Logo programming is illustrated in the sequences in grades 4-12, it is clear for Bitter that the language of choice at older levels is BASIC or PASCAL.

However, it needn't be the case that Logo use is restricted to, or even necessarily started with, students at a very young age. Successful Logo use projects have been introduced for division II students (e.g., Richardson & Dobson [1984*], Cathcart [1984*], Knoepfell [1984*] and division III students (e.g., the original use of Logo in Boston (Bitter & Watson [1983], and Morgan [1983] here). Moore (1984*) teaches Logo in a first course in a sequence--Computer Science for teachers--at Oregon State. She sees Logo as an ideal way to focus on such ideas as various uses of words in computing, structured programming, driver procedures, local versus global variables and recursion (in fact, her 1984 book Activities for Logo Discovery contains perhaps the best introduction to recursion in a list processing sense). Moore finds that the Logo work enables her students to make much easier and more rapid progress in a later study of Pascal. Allan (1984) sees Logo as a valuable way into the knowing of the artificial intelligence language LISP.

Because of the levelled nature of Logo, it is accessible (and was designed to be so) by persons at many developmental levels. With younger (division 1 and 2) children, most users contacted advised against a fast multilevelled introduction, but rather one which responded to student's own needs (Krasner & Mitterer, 1984). Premature prompting of students to use procedures, sub-procedures, variables, recursion, or top-down programming do not seem fruitful. On the other hand, there are division II aged students who can do sophisticated Logo procedures. There is a reasonably large and growing bank of resources (see appendix) which would enable a teacher to provide students with activities which focus on any level of Logo use and which provide what appear to be interesting tasks to be accomplished at that level. These activities highlight the young person controlling and designing procedures to control various "robotic" devices (turtle, floor turtle, poetry maker, music maker, etc.). In fact, many educators appear to favor the use of cheap programmable robots such as Big Trac as a prelude to Logo use (e.g., Richardson [1984*], Sawada [1984*] here, D.C. Johnson [1984*] at Chelsea College in London). Another aspect of literacy emphasized in Logo use is the control of location/motion on a screen, as well as the notion of a shape and shape-table (in Sprite use).

To summarize, Logo can and is successfully used as a way into robotics and computer literacy and science. There is debate as to its most effective level of introduction and use. Although micro-computer Logo can allow for controlling many robotic environments, it has not been designed as a quantitatively oriented language. Thus, there are many computer uses and applications for which it is unsuited. However, contrary to the points of view of some, Logo has been effectively introduced and used at a variety of ages for literacy purposes. It appears that the most effective of these uses were for longer (months, full year), rather than shorter (a few sessions) time periods.

3.5 Mathematics

In Mindstorms, Papert claims that Turtle Geometry under Logo is ego-syntonic mathematics (that is, mathematics which develops in concert with normal human thinking.) Thus, one might expect Logo to be used widely in mathematics. Such use is, of course, mainly seen in geometry. Such use is deep, ranging from shape and orientation work with young children through a sophisticated university course, Turtle Geometry (Abelson & DiSessa, 1981) at MIT. This geometry is a local geometry and allows for experience with geometric objects from direct actions, through properties, relationships and logic (Kieren & Olson, 1983). This geometry is also one which intimately relates shape, motion and number.

There is considerable good material available here as well. Moore (1984a) has a book Geometric Experiences for Logo Discovery, which relates well to the geometry curriculum of Alberta in divisions II and III. In addition, it contains material which can allow for naive insights into artificial intelligence and geometry. There are numerous excellent procedures in Abelson (1982a, 1982b), Watt (1983), and Thornburg (1983) which allow a student to explore the nature of shape, construction and motion in sophisticated ways. There is also excellent work on geometry and nature, as well as tasks exploring dimensionality (Thornburg, 1983).

Logo packages can be found which allow young children to directly explore geometric ideas such as shape and angle through single key presses (e.g., Instant Logo). Work with these young children seems to have been quite effective (Dobson & Richardson, 1984*, Cathcart & Cathcart, 1984*, Berdonneau, 1984*).

One can create a three-dimensional representation utility using Logo to define single key procedures which allow such perspective drawing (Olson

[1984*], Kieren, 1984*). Thus, division II and III students can use such a tool to work on representations of three dimensions in two, a useful skill, but more importantly an activity which helps geometric thinking. Because students can look at (and edit) the procedures in such a utility, the students can also build their own computer "language," a very significant literacy/ science outcome.

There is less published number oriented work in Logo. However, there has been a correlation between Logo experiences and non-geometric curriculum objectives as well. Harrison and Shroder (1984*), at the University of Calgary, have described uses in number theory and probability.

3.6 Art

Logo presents an interesting medium for early work in graphic art. One can write procedures which have rather dramatic visual effects. It is, however, an art which is generated through rather organized mental and linguistic activity. While many of the "artistic" representations in Logo (houses, flowers) are almost cliches, there are a very wide range of possibilities. One such is the construction of a Logo quilt (Moore, 1984b). Another involves middle eastern "rug" art (Burnett, 1984*). Burnett (1984*) also has done very interesting work on graphics of various "play" environments, particularly those of model railroad track schemes which represent very interesting design problems.

There are those who would argue that art activities should involve physical work and work which is less structured. Logo art emphasizes mental organization and control over "art making" devices.

3.7 Language

As noted at the outset, Logo is a qualitative, linguistically built language. Its first application came in a linguistic situation: creating pig-latin. Because Logo is Lisp-like, it has list processing primitives. These primitives allow one to write procedures which define a certain grammatical, syntactic structure. Students can either work with a teacher made structure/procedure to create letters, poems or even stories, or design their own procedures, and thus syntactic structures. One can also study the artificial products of such programs for their language usage and patterns. There are a number of very nice commercial materials in this area as well (Watt [1983], DeWitt [1984]).

3.8 Social Studies

There is a very interesting Logo procedure done by Deleo (Schroder & Harrison, 1984*) at the University of Calgary which allows students to study in detail the effect of scale change on maps and their interpretation. Students can use this utility to study and predict what happens to representations when a scale change is made. They also study problems where they predict needed scale changes.

3.9 Music

Music was one of the original applications of Logo use (Bamberger, 1982). Logo is used to create scores for various machine controlled instruments. It can also be used to generate microcomputer music (Campbell, Klassen, & Glaser, 1983). One of the most interesting versions in this regard is Commodore '64 Logo, which has primitives which control the SID chip. Writing procedures for music provides a focus on the rhythmic and structural aspects of music.

3.10 Data Structures

Because of its list processing capability and structure, Logo provides a sophisticated introduction to how various file management systems can be designed and manipulated. For example, Allan (1984) gives a Logo example of list which can be used for designing a mailing list. Part of this example is shown below:

```
MAKE "STANDARD.INFO [: NAME :ADDRESS :PHONE]
```

Since each of the 3 elements of this list is in itself a list, the postal code of a person could be abstracted from the overall list by the following Logo command:

```
MAKE "POST.CODE LAST :ADDRESS
```

The LAST primitive takes out the last element from the list :ADDRESS.

3.11 Problem Solving

In a time when problem solving is a touch phrase in education, use of Logo has been thought of as fostering such behavior in children. Indeed, Logo use seems to drive many activities associated with problem solving: pre-planning or structuring procedures, matching a language sequence with an action sequence to reach a goal, debugging a procedure to better approximate a goal (Krasner & Mitterer, 1984). It is fair to say that most Logo activities found in the literature or curriculum materials are cast in the form of a goal state to be reached and a set of given procedures to be used or procedures to be written to have a computer generate the goal. Even micro-worlds as discussed by Papert (1979, 1980), Lawler (1983), Abelson (1982), Higginson (1982) and others conceive of a child or other person placed in an environment with certain available procedures. The person explores this micro-world by defining problems or tasks to work on. Thus, in all of its uses Logo is seen explicitly or implicitly as a problem solving task.

The instructional structures for such tasks vary greatly from situation to situation. Some teacher/users of Logo, while providing task cards for students to work on, see student free selection of problems as very important (e.g., Richardson & Dobson, 1984*). Other teachers think students working individually or in groups should work on interesting problems (mathematical, computer or other) which are set for them (e.g., Gloria Cathcart, 1984*). Others see Logo use as working through a sequence of task cards, each of which sets a problem which augments the student's current knowledge of Logo (e.g., Knoepfell, 1984*) or as working selectively through a levelled set of problem tasks (e.g., Richards, 1984). Finally, there are many examples of students using given procedures and inventing others in an environment of a large general problem (e.g., model railroad layouts [Burnett, 1984*], designing model figures [Campbell, 1984*]).

3.12 Summary

Logo can be, and is being, used at a variety of levels to support a wide variety of school objectives. Its most popular use seems to be a problem solving/computer literacy use with division II students.

Mathematics, again at many levels for many purposes, seems to be another natural place for Logo use. In light of the Alberta division II and III mathematics curricula, using Logo as an important means of attaining geometry as well as logic and problem solving goals appears appropriate. Use of Logo in other curriculum areas assumes student use of teacher/commercial procedures or requires Logo and language use skills at a higher level.

3.13 Logo With Less Able Students

Logo has been used in a variety of ways with learning disabled students and with others whose school success record has not been good (Weir & Watt, 1983). Learning disabled students aged 8-12 using teacher-made procedure games, have worked on and improved orienting skills (Simon, 1984*). Weak 12-year-old students have worked with Logo in mathematics with the aim being to improve their sequential thinking skills in mathematics (Howe, O'Shea, & Plane, 1980). Finally, Mathematics 15 students here in Edmonton successfully learned Logo and seemed to benefit from the experience on the affective level (Wood, 1984). Krasner and Mitterer (1984) note, rightfully, that there is a paucity of evidence on the value of Logo use in this area, however.

4. Why Logo: The Word From Research

When one thinks of research on Logo use, two conflicting thoughts arise. On the one hand Logo is a carefully researched educational innovation. Because its genesis in artificial intelligence laboratories in the U.S. and Scotland occurred before widespread use of micros, the initial use of Logo was limited and carefully studied. If for no reason other than economic, Logo was not dumped in a premature form on schools. There are a large number of technical reports, each of which describe a use of Logo in a particular way with a select teacher and select student group (e.g., Feurzig, Papert et al. [1968], Goldstein & Miller [1976]).

On the other hand, there is little research until recently which has been done independently of persons/settings which were biased toward Logo use. Further, there has been little study of any large scale implementations or even implementations without rather selected teachers (Krasner & Mitterer, 1984).

4.1 Effects of Logo Use

Many of the MIT Logo reports give details of what kinds of procedural thinking children exhibited under different Logo use conditions. It appears that given sufficient adult support and attention many children of upper elementary school age can write reasonably sophisticated procedures. However, little research has been done to see if children will work at higher levels without adult support, provocation or modelling.

In a report most critical of Logo, Pea of Bank Street (e.g., Jewson & Pea, 1982) found that upper elementary Logo users did not use structured or procedural thinking in doing a non-Logo task, which would easily admit such thinking. This report seems to contradict the writing of Logo supporters. It should be noted, however, that the users in the Bank Street study had access to Logo in a very casual environment. These students were not instructed in any systematic way. Thus, one probably can best conclude that mere access to or playing with Logo does not guarantee any particular intellectual gains. This conclusion is backed by Krasner and Mitterer (1984), who see three conditions central to promoting problem solving through Logo use: teaching techniques which foster the general utilities of the thinking used in Logo; selection of children intellectually mature enough for the particular necessary Logo use; and inclusion of all components (structure, debugging) of the Logo environment because programming per se is not sufficient for transfer. Clements (1984), in an exploratory study of young grade 2-3 students using Logo in a more guided way, did find intellectual growth which seems attributable to Logo use.

There is mixed evidence of effects of Logo in mathematics. In work with low achievers in the U.K., Howe, O'Shea and Plane, 1980, did not find Logo users advancing in their mathematical abilities. However, Noss (1984) of the

Chiltern (U.K.) Project found that children aged 8 to 11, taught by teachers with only a limited amount of training in Logo, exhibited important mathematical thinking processes after using Logo about 75 minutes/week for a year. She saw universal instances of geometric generalizing, as well as a proclivity to conjecture and look for underlying rules and theorems. Berdonneau and Dumas (1981) in France found 11-year-olds naturally working on composition and reversibility in computation through Logo use. They saw Logo use as enhancing number skills rather than ignoring them as some critics might suggest.

In terms of problem solving, both Dobson and Richardson (1984*) and Gloria Cathcart (1984*) here in Edmonton report excellent student (aged 9-11) outcomes from a year of regular (50 or more minutes/week) Logo use. The former reported that regular class use of Logo seemed to transfer to other problem solving use in other classes. Cathcart's selected students solved interesting mathematics and computing problems even to the level of mention by the editors of The Computer Teacher journal.

4.2 On Language Use

Most Logo applications have involved two or more students working together at a micro either using pre-developed procedures or writing and applying their own. Under these circumstances students have been encouraged to discuss their thinking and reactions with one another. One finding which seems almost universal in the research on Logo use, even in those studies generally critical of Logo, is that students of nearly all ages eight through university level, seem to be able to talk about their reasoning in intelligent ways. For example, Bideault-Delevanne (1983) in France found 8- to 10-year-old children able to speak much more precisely about measures and distances than students doing similar tasks without Logo. Children seem aware of how their ideas work and can discuss this clearly. Critics of computer use point to isolation and lack of communication as serious problems with computer use. This does not seem to be a Logo associated problem. In fact, most research has found Logo use is promoting a social/intellectual environment.

Another criticism of Logo language use is that it fosters a limited "commanding demanding" use of language (Burch [1984], Sullivan [1983]). While adherents of Logo do not see this as a problem, there has been no study relating Logo use to language used by its users in general.

4.3 Teacher Education

There seems to be a unanimity among persons interviewed and literature studied that teacher education is a critical feature in the effectiveness of Logo use. There is some research showing that novice teachers can learn Logo in a reasonably short period of time and can use these ideas in teaching (Ferres [1983] at Oregon). However, Ferres' success was not universal to all teachers he worked with.

Moursund (1983-84) fears that without proper training and insight teachers will accept minimal Logo work as real computer literacy, that they will not be aware of the complexities of Logo, nor will they implement its problem solving aspects. The problem is getting the "ordinary" teacher to do what the "expert" teacher has done with Logo.

One key feature of successful teacher education programs in Logo seemed to be the task of designing Logo units and being able to try them critically with children (Westrom & Egnatoff at U.B.C. [1984*], Burnett & Higginson at Queens [1984*], Kieran at UQAM [1984*]). Another important feature is training in Logo classroom management. Most successful Logo teacher/users claim to use indirect rather than direct teaching methods (Noss [1984], Berdonneau [1984*]). Usually children (or adults) in Logo classes are working on different tasks and by the nature of the language at different levels. This usually puts the teacher in the role of an advisor rather than "teller." For example, Kieran (1984*) notes that in Logo one must allow students to do mathematics rather than merely learn about it.

4.4 Cost

Logo has been used in a wide variety of physical circumstances. Some users have one or two computers for individual or whole-class use in a stand-alone classroom. Other users bring students to a laboratory to work on the computers (with 2 or 3 students per computer). In nearly all places observed or read about, one main Logo language disk set was used.[4] Further, most teachers seemed to think it important for any student (or student group) who was writing procedures (usually division II aged and up), to have their own initialized disk on which to store their own (and given) procedures. Thus, the cost of Logo use seems to be that of a Logo package (Apple, Terrapin, Commodore, etc.) at about $150 per school plus one disk per person or user group ($3-$5 each).

Thus, the cost of adding Logo (provided one has 64K computers) to an instructional program for 200 students, for example, would be $150 plus $300 to $600 for disks depending on price and number. (The cost per student beyond computer hardware costs is between $3 and $5 per student per year).

---

[4] This is not without its problems. A teacher booting Logo on a number of machines from one disk faces time and logistic problems. This seems particularly important/difficult for less experienced teachers or with younger users (Higginson, 1984*).

5. Recommendations

I. Logo use is worth continued encouragement and study in Alberta schools.

Logo is already being used in a large number of Alberta elementary schools and a few secondary schools. Since Logo use appears to have a variety of positive effects, such use can continue to be encouraged. On the other hand, much research is needed to ascertain what the effects of various Logo uses are and what effective programs of use are like.

II. The role of Logo in computer literacy bears study.

Logo will not replace the use of other languages and computer utilities. It has characteristics which make it an attractive component of computer literacy and computer science programs. Who can learn appropriately from various Logo experiences and how this is best done needs serious research.

III. Logo is useful in studying robotics.

In an age where "working" may come more and more to mean adjusting, setting or designing robot uses, Logo is a general robot control language and is useful for our students to consider.

IV. Logo use should be extensive instead of short term.

Because of its levelled nature, Logo use can start very quickly. Yet sophistication in use develops slowly and appears to be worth developing. This sophistication will not best be served by a rigid, lockstep, shallow use of Logo.

V. Logo can be used at any level in the Alberta schools.

Because Logo is a deep language and can be looked at in a number of curricular ways (e.g., turtle graphics, geometry, list processing, word processing). Its use can be appropriate to many different kinds of students for many different purposes.

VI. Logo use in division II and III mathematics curriculums needs investigation.

It appears that a profound change in the curriculum, particularly geometry (but also logic and mathematical problem solving), would arise from Logo use. Just how the curriculum (from objectives to construction of materials) might change, and the effects of such change, need study.

VII. Logo as a vehicle for problem solving needs clearer definition and research.

As problem solving is a current thrust in education and widely mentioned with Logo, such research is indicated. This is especially so given the paucity of work done to date.

VIII. The relationship between various styles of Logo use and intellectual communication and social relationships among students should be studied.

In an age where computers are seen by some to be depersonalizing, Logo use seems to promote certain verbal skills and communication skills in its users. This bears further study with the point being to better understand humane computer uses in general.

IX. If one wishes to use Logo, there are a large number of reasonable materials available. Use should be made of these.

For many different uses at many levels, there are materials available. Many of these materials would be best aimed at teachers, but there are student materials as well. A good number of these are Canadian in origin.

X. Rather than material development, curriculum development efforts should focus on what goals and objectives or Logo use are appropriate and how these can be achieved.

XI. Logo use needs a very effective scheme of teacher education.

Because of the nature of the language, the variety of its possible uses, and the instructional style of its use, Logo demands rather extensive teacher education. This must include language study, curriculum study, instructional practice study and the chance to implement and evaluate appropriate Logo uses.

XII. A wide variety of research/development projects with Logo are indicated.

There is a growing body of research on Logo use which is becoming available. Still such efforts are small in number. Because using Logo is no singular thing, a variety of research is posited as well. Because there is no such thing as "standard Logo use," uses designed particularly for Alberta curriculums will need research.

XIII. Research and development projects will be best fostered by a number of small projects rather than a single large project.

This follows from the need to study various levels of Logo use as well as the effects of varieties of such use. It would, however, be important to coordinate the design but more importantly the compilation, distillation and dissemination of results from such developments and research.

APPENDIX I

Selected Reference Bibliography

Abelson, H. (1982a). Apple LOGO. Peterborough, NH: BYTE/McGraw-Hill.

Abelson, H. (1982b). LOGO for the Apple II. Peterborough, NH: BYTE/McGraw-Hill.

Abelson, H. (1982, August). A beginner's guide to LOGO. BYTE, 88-112.

Abelson, H., & DiSessa, A. (1981). Turtle geometry: The computer as a medium for exploring mathematics. Cambridge, MA: Massachusetts Institute of Technology (MIT) Press.

Allan, B. (1984). Logo. London: Pitman.

Allen, J. (1979, August). An overview of LISP. BYTE, 96-100.

Austin, H. (1976). Teaching teachers LOGO: The Lesley experiments. (LOGO Memo No. 23). Cambridge, MA: MIT Artificial Intelligence Laboratory.

Bamberger, J. (1979). LOGO music projects: Experiments in music and design. (LOGO Memo No. 52). Cambridge, MA: MIT Artificial Intelligence Laboratory.

Bamberger, J. (1982, August). LOGO music. BYTE, 325-328.

Bandelier, N. (1982, November). TI LOGO and first-graders--a winning combination. The Computing Teacher, 38-39.

Barnes, B. J., & Hill, S. (1983, May). Should young children work with microcomputing--Logo before Lego? The Computing Teacher, 11-14.

Bearden, D. (1983-84, December/January). Optical illusions--a challenge in problem solving. The Computing Teacher, 75-77.

Bearden, D., & Muller, J. (1983, April). Turtle Graphics--on and off the screen. Classroom Computer News, 33-34.

Bearden, D., Muller, J., & Martin, K. (1982). The Turtle's sourcebook. Richardson, TX: Young Peoples' LOGO Association.

Beasley, W. A. (1983, October). Printing LOGO graphics. Creative Computing, 262-264.

Berdonneau, C., & Dumas, R.-M. (1981). Une tortue dans la classe. Compte-rendu d'experimentation en classe de Cours Moyen deuxieme annee IREM, Paris Nord.

Bert, C. (1983). Logo a l'ecole: Reflexions d'une observatrice. Actes du 1er Colloque Logo. Clermont-Ferrand, France.

Bideault-Delavenne, A. (1983). L'enfant et l'ordinateur: strategies de construction de parcours en milieu Logo. Unpublished document. France.

Billstein, R. (1982, November). Learning LOGO and liking it. The Computing Teacher, 18-20.

Billstein, R., & Moore, M. L. (1983-84, December/January). Recursion, recursion. The Computing Teacher, 46-47.

Birch, L. (1983, April). The Turtle's corner. Classroom Computing News, 29-32.

Bitter, G. (1982-83, September/February). The road to computer literacy. Electronic Learning. 6-part series.

Bitter, G. G., & Watson, N. R. (1983). Apple Logo primer. Reston, VA: Reston Publishing.

Bitter, G., & Camuse, R. A. (1984). Using a computer in the classroom. Reston, VA: Reston Publishing.

Boecker, H.-D., & Fischer, G. (1982, September). Logo project Prokop. BYTE, 329-330.

du Boulay, B. (1977). Learning teaching mathematics. Paper presented to "Computers in Higher Education" Conference of the Centre for Advancement of Mathematical Education in Technology.

du Boulay, B., & Howe, J. (1981a). Relearning mathematics through Logo: Helping student teachers who don't understand mathematics. In J. A. M. Howe & P. M. Ross (Eds.), Microcomputers in secondary education: Issues and techniques (pp. 69-81). London: Kogan Page.

du Boulay, J. B. H., & Howe, J. A. M. (1981b). Student teachers' attitudes to maths: Differential effects of a computer-based course. In R. Lewis & D. Tagg (Eds.), Computers in education. Dordrecht: North-Holland Publishing.

Bull, G. (1982, November). Microworlds. The Computing Teacher, 62-63.

Bull, G., & Tipps, S. (1983-84, December/January). Problem spaces in a project-oriented Logo environment. The Computing Teacher, 54-57.

Burch, R. (1984). Technology and Curriculum: Toward a Philosophical Perspective. Occasional Paper 27, Curriculum Praxis. Edmonton: Department of Secondary Education, University of Alberta.

Burnett, J. D. (1982). LOGO: An introduction. Morris Plains, NJ: Creative Computing Press.

Burrowes, S. (1983, April). Some Logo drawing ideas. The Computing Teacher, 64-65.

Campbell, S., Klassen, W., & Glaser A. (1983). Elementary Logo module. Richmond, BC: Richmond School District.

Carlson, E. (1982, October). The three faces of Apple Logo. Micro, 32-35.

Carter, R. (1981, September). Logo and the Great Debate. Kilobaud Microcomputing, 48-51.

Carter, R. (1983, April). The complete guide to Logo. Classroom Computer News, 35-39.

Ciarcia, S. (1982, August). High-resolution sprite-oriented color graphics. BYTE, 57-80.

Clements, D. H. (1983-84, December/January). Supporting young children's Logo programming. The Computing Teacher, 24-30.

Coon, M. (1982, November). Papert at the faire. The Computing Teacher, 14-16.

Crandall, N. D. (1977). Computer-assisted instruction: How it raises children's achievement scores. Internal report. Los Nietos S.D., CA.

DiSessa, A., & White, B. (1982, August). Learning physics from a Dynaturtle. BYTE, 324.

Eyster, R. H. (1981, December). Seymour Papert and the Logo universe. Creative Computing, 70.

Fagan, E. R. (1983, April). Writing/reading: Logo's syntonic learning. The Computing Teacher, 43-45.

Ferres, G. W. (1983). Training and implementation strategies appropriate to the introduction of Logo into teachers' curriculum. Unpublished doctoral dissertation, University of Oregon, Eugene, OR.

Feurzeig, W., & Lukas, G. (1972, March). LOGO--a programming language for teaching mathematics. Educational Technology, 39.

Feurzeig, M., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1968). Programming languages as a conceptual framework for teaching mathematics. Boston, MA: Bolt, Beranek & Newman.

Frye, N. (1982). The great code. Toronto: Academic Press.

Goldenberg, E. P. (1979). Special technology for special children. Baltimore, MD: University Park Press.

Goldenberg, E. P. (1982, August). Logo--a cultural glossary. BYTE, 210-228.

Goldstein, I., & Miller, M. (1976). AI-based personal learning environments: Directions for long-term research. (Logo Memo No. 31.) Cambridge, MA: MIT Artificial Intelligence Laboratory.

Gorman, H., Jr. (1982, August). The Lamplighter Project. BYTE, 331-332.

Harrison, D. B. (1983). Logo microcomputer projects to complement junior high mathematics topics. Paper presented at MCATA, Calgary.

Harvey, B. (1982, August). Why Logo? BYTE, 163-193.

Higginson, W. (1982, August). Leading fish to water: Early observations on the use of Logo. BYTE, 328-329.

Higginson, W. (1982). Leading trout to water: Observations on the use of Logo by able children. In Proceedings of the Sixth Annual Meeting of the International Group for the Psychology of Mathematics Education, Antwerp.

Howe, J., & O'Shea, T. (1976). Computational metaphors for children. Paper presented at the XXIst International Congress of Psychology, Paris.

Howe, J., & O'Shea, T. (1978, January). Learning mathematics through LOGO. ACM Sigcue Bulletin, 12(1), 2-11.

Howe, J., & O'Shea, T., & Plane, F. (1980). Teaching mathematics through Logo programming: An evaluation study. In D. Tagg & R. Lewis (Eds.), Computer-assisted learning: Scope, progress and limits. Dordrecht: North-Holland Publishing.

Howe, J., & Ross, P. (Eds.). (1981). Moving LOGO into the mathematics classroom. Microcomputers in secondary education: Issues and techniques (pp. 89-101). London: Kogan Page.

Hunter, B. (1983). My students use computers. Reston, VA: Reston Publishing.

Jewson, J., & Pea, R. D. (1982, August). Logo research at Bank Street College. BYTE, 332-333.

Kelman, P. (1981, March/April). Journey through Mathland: An interview with Seymour Papert. Classroom Computer News, 10-11.

Kelman, P. (1983, April). Seymour, has your dream come true? Classroom Computer News, 7.

Kelman, P., et al. (1983). Computers in teaching mathematics. Boston, MA: Addison-Wesley.

Kieren, T., & Olson, A. (1983). Linking Logo and learning levels. Paper presented at the Northwest Computing Conference, Corvallis, OR.

Kleiman, R. B. (1982, July). Teaching Johnny to program. Compute!, 16-18.

Krasner, L. R., & Mitterer, J. G. (1984). Logo and the development of general problem solving skills. Alberta Journal of Educational Research, XXX(2).

Kulik, J. A., Bangert, R. L., & Williams, G. W. (1983, February). Effects of computer-based teaching on secondary school students. Journal of Educational Psychology, 216-224.

Lawler R. W. (1976). One child's learning: Introducing writing with a computer. (Logo Memo No. 56.) Cambridge, MA: MIT Artificial Intelligence Laboratory.

Lawler R. W. (1982, August). Designing computer-based microworlds. BYTE, 138-160.

Lawler, R. W. (1982, September). Logo ideas. Creative Computing, 126.

Lawler, R. W. (1983, January). Logo ideas: Microworlds and learning/The idea of a formalism. Creative Computing, 132.

Lemmons, P. (1982, August). Logo update. BYTE, 334-340.

Leron, U. (1982, August). The group of the turtle. BYTE, 330-331.

Lough, T. (1982, November). Watercross: A Logo exploration. The Computing Teacher, 51-54.

Lough, T. (1983, March). Logo--Discovery learning with the classroom's newest pet. Electronic Learning, 49-53.

Lough, T. (1983, April). Exploring new horizons with Logo. Electronic Learning, 71-75.

Lough, T. (1983-84a, December/January). Where is Logo? The Computing Teacher, 30-32.

Lough, T. (1983-84b, December/January). A cure for recursion. The Computing Teacher, 34-37.

Lough, T., & Tipps, S. (1983, April). Is there Logo after turtle graphics? Classroom Computer News, 50-53.

Mace, S. (1983, January 23). Where is Logo taking our kids? Infoworld.

Maddox, A. (1984). Teaching computing science: An analysis of the Ontario Computing Science curriculum. Unpublished paper, Department of Secondary Education, University of Alberta.

Markuson, C., Tobias, J., & Watt, D. (1983, January). Logo fever. Instructor.

Martin, K., Bearden, D., & Muller, J. (1982, November). Turtle graphics on and off the computer. The Computing Teacher, 55-58.

Martin, K., & Berner, A. (1982, November). Teaching turtles. The Computing Teacher, 41-45.

Martin, K., & Riordon, T. (1983, October). Big kids helping little kids. The Computing Teacher, 30-32.

Martin, K., & Riordon, T. (1983-84, December/January). Poemprint, Piglatin, Madlibs. The Computing Teacher, 70-75.

Martin, K., & Riordon, T. (1984, February). Polyspirals. The Computing Teacher, 53-55.

Martin, K., & Riordon, T. (1984a, May). So what about sprites? The Computing Teacher, 63-66.

Martin, K., & Riordon, T. (1984b, May). More about sprites (and the turtle too). The Computing Teacher, 66-69.

McCauley, J. (1983-84, December/January). Kepler. The Computing Teacher, 15-22.

McCauley, J. (1984, May). Tree. The Computing Teacher, 15-22.

McClees, J. (1982, March). Where do turtles come from? Kilobaud Microcomputing, 99-100.

McDougall, A., Adams, T., & Adams, P. (1982). Learning LOGO on the Apple II. Sydney, Australia: Prentice Hall of Australia.

Moore, M. L. (1983-84, December/January). A recursion excursion with a surprising discovery. The Computing Teacher, 49-52.

Moore, M. L. (1984a). Geometry problems for Logo discoveries. Palo Alto, CA: Creative Publications.

Moore, M. L. (1984b). Logo discoveries. Palo Alto, CA: Creative Publications.

Moore, M. L. (1983, April). The art of teaching Logo. Classroom Computer News, 28-29.

Morgan, J. (1983). Logo and motion geometry: A study of a grade 7 classrom. A thesis proposal. Department of Secondary Education, University of Alberta, Edmonton.

Moursund, D. (1983-84, December/January). Logo frightens me. The Computing Teacher, 3-4.

Muller, J. (1982, August). Young People's LOGO Association. BYTE, 333-334.

Muller, J. (1982, October). The friendly computer language. Creative Computing, 55-60.

Muller, J. (1983-84, December/January). Dumping the turtle. The Computing Teacher, 79-81.

Nelson, H. (1981a, Summer). Seymour Papert: Spearheading the computer revolution. On Computing, 10-12.

Nelson, H. (1981b, Summer). Learning with LOGO. On Computing, 14-16.

Nelson, H. (1981, June). Logo for personal computers. BYTE, 36-44.

Nelson, H. (1982a, March). Logo: Not just for kids. Kilobaud Microcomputing, 96-107.

Nelson, H. (1982b, March). Dynaturtle (Turtle graphics and Newtonian Physics). Kilobaud Microcomputing, 104-105.

Noss, R. (1984). Children learning LOGO programming. Interim Department No. 2 of the Chiltern Logo Project. Hatfield, England: Advisory Unit for Computer-based Education.

Olds, F. (1984). CyberLOGO Turtle. Classroom Computer News, 60-63. (Review of limited turtle graphics program.)

Overall, T., Lola, P., Murphy, K., Dafoe, M. M., Housey, J., Leventhal, S. (1981, September). Learning with Logo at the Lamplighter School. Kilobaud Microcomputing, 42-47.

Papert, S. (1971). Teaching children to be mathematicians vs teaching children about mathematics. (Logo Memo No. 4). Cambridge, MA: MIT Artificial Intelligence Laboratory.

Papert, S. (1971). Teaching children thinking. (Artificial Intelligence Memo No. 247.) Cambridge, MA: MIT Artificial Intelligence Laboratory.

Papert, S. et al. (1979). The Brookline LOGO Project, final report. Part II: Project summary and data analysis. (LOGO Memo No. 53, Artificial Intelligence Memo No. 545.) Cambridge, MA: MIT Artificial Intelligence Laboratory.

Papert, S. (1980). Mindstorms: Children, computers and powerful ideas. New York: Basic Books.

Papert, S. (1980, September). New cultures from new technologies. BYTE, 103-105.

Papert, S. (1981, March). Computers and computer cultures. Creative Computing, 46-49.

Proud, J. (1983). A Logo in-service for educators. Unpublished master's thesis, California State Polytechnic University, Pomona, CA.

Riordon, T. (1982, November). Creating a LOGO environment. The Computing Teacher, 46-50.

Riordon, T. (1983-84, December/January). Helping students with recursion: Teaching strategies. Part I: Introducing recursion with non-computer recursive experiences. The Computing Teacher, 38-42.

Riordon, T. (1984, February). Helping students with recursion: Teaching strategies. Part II: Moving to the computer. The Computing Teacher, 59-64.

Riordon, T. (1984, March). Helping students with recursion: Teaching strategies. Part III: Teaching students about embedded recursion. The Computing Teacher, 64-69.

Ross, P. (1983). Introducing LOGO. Boston, MA: Addison-Wesley.

Rousseau, J. F., & Smith, S. M. (1981, September). Whither goes the turtle? Kilobaud Microcomputing, 52-55.

Rowe, N. (1981). Grammar as a programming language. (Logo Memo No. 39.) Cambridge, MA: MIT Artificial Intelligence Laboratory.

Ryba, K. A., & Chapman, J. W. (1983, August). Toward improving learning strategies and personal adjustment with computers. The Computing Teacher, 48-53.

Sharples, M. (1981). A computer based teaching scheme for creative writing. In R. Lewis & W. Tagg (Eds.), Computers in education. Dordrecht: North-Holland Publishing.

Schneiderman, B. (1980). Software psychology: Human factors in computer and information systems. Boston, MA: Little, Brown.

Solomon, C. J. (1981). Apple Logo: Introduction to programming through Turtle Graphics. Lachine, PQ: LCSI.

Solomon, C. J. (1982, August). Introducing Logo to children. BYTE, 196-208.

Solomon, C. J., & Papert, S. (1976). A case study of a young child doing turtle graphics in LOGO. (Artificial Intelligence Memo No. 375.) Cambridge, MA: MIT Artificial Intelligence Laboratory.

Staples, B. (1981, December). Turtles and sprites for the 99/4. Creative Computing, 45-48.

Stavely, T. (1982, December). List processing in Logo. The Computing Teacher, 43-47.

Steffin, S. A. (1983, April). The instructional applicability of Logo. Classroom Computer News, 88.

Stone, G. (1983, April). Solving problems with Logo. In Cider, 50-55.

Sullivan, E. V. (1983). Computers, culture and educational futures: A critical, reflective meditation on Papert's "Mindstorms". Unpublished manuscript.

Swett, S. (1983, May/June). Logo offspring: A look at modified versions of Logo and turtle graphics programs. Electronic Learning, 72-75.

Therrien, D. (1984). Micro-Maths: comment illustrer des concepts mathematiques en l'aide du micro-ordinateur. Montreal, Les Editions Ville-Marie.

Thornburg, D. (1981, March). Logo and Pilot languages. Compute!, 14-18.

Thornburg, D. (1982). Every kid's first book of robots and computers. Greensboro, NC: Compute Books.

Thornburg, D. (1983). Discovering Apple Logo. Boston, MA: Addison-Wesley.

Torgerson, S. (1983-84, December/January). Classroom management for Logo. The Computing Teacher, 12-14.

Torgerson, S., Kriley, M. K., & Stone, J. (1983). Logo in the classroom. Eugene, OR: International Council for Computers in Education.

Upitis, R. (1982a, November). Turtle talk: An annotated selection of Logo programs. The Computing Teacher, 22-27.

Upitis, R. (1982b, November). Logo and the primary-junior pupil: One student's first encounter. The Computing Teacher, 28-31.

Voderberg, H. (1984, March). Use Logo graphics in your BASIC program. The Computing Teacher, 10-11.

Watt, D. (1979). Final report of the Brookline LOGO projects. Part III: Profiles of individual students' work. (Artificial Intelligence Memo No. 546). Cambridge, MA: MIT Artificial Intelligence Laboratory. (ERIC Document Reproduction Service No. ED 207 802).

Watt, D. (1979, December). A comparison of the problem solving styles of two students learning LOGO: A computer language for children. Creative Computing, 86-90.

Watt, D. (1982, July). Teaching Turtles: LOGO as an environment for learning. Popular Computing, 48.

Watt, D. (1982, August). Logo in the schools. BYTE, 116-134.

Watt, D. (1982, September). Should children be computer programmers? Popular Computing, 130-133.

Watt, D. (1983, April). Learning with LOGO. Classroom Computer News, 40-43.

Watt, D. (1983). Learning with LOGO. New York, NY: McGraw-Hill.

Watt, D. (1983, August). Logo: What makes it exciting? Popular Computing, 106-166.

Watt, M. (1982, October). What is Logo? Creative Computing, 112-129.

Watt, M. (1984, February). De-bug collection. In Cider, 78-84.

Watt, M. (1984, March). Logo: Where's the pony? In Cider, 86-89.

Webb, G. (1982, May). Beyond BASIC: Powerful ideas and LOGO. The ATA Magazine, 16-19.

Weinreb, W. (1982, November). Problem solving with Logo. BYTE, 118-134.

Weintraub, H. (1983, September). Putting Logo to work. The Computing Teacher, 52-54.

Weir, S. (1981, September). Logo and the exceptional child. Kilobaud Microcomputing, 76-84.

Weir, S., & Emanuel, R. (1976). Using Logo to catalyse communication in an autistic child. (Department of Artificial Intelligence Report No. 15.) University of Edinburgh.

Weir, S., Russell, S. J., & Valente, J. (1982, September). Logo: An approach to educating disabled children. BYTE, 76-84.

Weir, S., & Watt, D. (1983, January). Logo: A computer environment for learning-disabled students. The Computing Teacher, 11-19.

Weizenbaum, J. (1976). Computer power and human reason: From judgment to calculation. San Francisco, CA: W. H. Freeman, 111-131.

Wierzbicki, B. (1984, January 23). Success of Logo hinges on teacher training. Infoworld.

Williams, G. (1982, August). Logo for the Apple II, the TI 99/4A and the TRS-80 color computer. BYTE, 230-290.

Wold, A. L. (1983, April). What is a programming language? Classroom Computer News, 46-49.

Wood, P. H. (1984). LOGO programming in Mathematics 15. Colloquium monograph. Department of Secondary Education, University of Alberta, Edmonton.

APPENDIX II

Curriculum Materials

Abelson, H. (1982a). Apple Logo. Peterborough, NH: BYTE/McGraw-Hill.

Abelson, H. (1982b). LOGO for the Apple II. Peterborough, NH: BYTE/McGraw-Hill.

Abelson, H., & DiSessa, A. (1981). Turtle geometry: The computer as a medium for exploring mathematics. Cambridge, MA: MIT Press.

Bearden, D. (1983). 1, 2, 3, my computer and me: A LOGO funbook for kids. Reston, VA: Reston Publishing.

Bearden, D., Martin, K., & Muller, J. (1983). Turtle's sourcebook (2nd ed.). Reston, VA: Reston.

Bitter, G. G., & Watson, N. R. (1983). Apple Logo primer. Reston, VA: Reston.

Burnett, J. D. (1981). LOGO: An introduction. Morris Plains, NJ: Creative Computing Press.

Cathcart, G. M., & Cathcart, W. G. (1984). Learning about computers. Toronto: Gage.

Davis, J., & Schwenker, J. (1982). LOGO: A problem solving approach. Spokane, WA: Turtle Enterprises.

Dewitt, S. (1984). Apple Logo activities. Reston, VA: Reston Publishing (A Prentice-Hall Company).

Dobson, J., & Richardson, P. (1984). Logo activities: A problem solving approach. Don Mills: Addison-Wesley.

Green, C., & Jaeger, C. (1984). Teacher, kids and LOGO. Irvine, DA: EduComp Publications.

Hunter, B. (1983). My students use computers: Learning activities for computer literacy. Reston, VA: Reston.

Kelman, P. et al. (1983). Computers in teaching mathematics. Reading, MA: Addison-Wesley.

MECC. (1983). Apple LOGO in the classroom. St. Paul, MN: Minnesota Educational Computing Consortium.

Miller, S., & Thorkildsen, R. (1983). Getting started with LOGO. Developmental Learning Materials.

Moore, M. L. (1984a). Geometry problems for LOGO discoveries: Explorations in turtle geometry. Palo Alto, CA: Creative Publications.

Moore, M. L. (1984b). LOGO discoveries: Explorations and programming activities for beginners. Palo Alto, CA: Creative Publications.

Papert, S. (1980). Mindstorms: Children, computers and powerful ideas. New York: Basic Books.

Richard, G. (1984). Logo activity cards. Edmonton: Access.

Ross, P. (1983). Introducing LOGO. Reading, MA: Addison-Wesley.

Solomon, C. J. (198). Apple Logo: Introduction to programming through Turtle Graphics. Lachine, PQ: LCSI.

Thornburg, D. D. (1983). Discovering Apple LOGO: An invitation to the art and pattern of nature. Reading, MA: Addison-Wesley.

Torgerson, S. (1983). LOGO lessons: Ideas for the classroom. Eugene, OR: International Council for Computers in Education.

Watt, D. (1983). Learning with LOGO. Toronto: McGraw-Hill Ryerson.

N.L.C. - B.N.C.
3 3286 05327106 6